LAWRENCE LIVERMORE NATIONAL LABORATORY

UCRL-TR-221841

Some experience with a H1-based auxiliary space AMG for H(curl)-problems

Tz. V. Kolev, P. S. Vassilevski

June 5, 2006

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

SOME EXPERIENCE WITH A H^1–BASED AUXILIARY SPACE AMG FOR $\boldsymbol{H}(\mathrm{curl})$–PROBLEMS

TZANIO V. KOLEV AND PANAYOT S. VASSILEVSKI

Abstract. This report provides several variants for constructing unstructured mesh AMG preconditioners for $\boldsymbol{H}(\mathrm{curl})$–problems exploiting $\boldsymbol{H}^1_0$–equivalent forms. The respective variants are illustrated with extensive numerical tests.

1. Introduction

The search for efficient preconditioners for $\boldsymbol{H}(\mathrm{curl})$ problems on unstructured meshes has intensified in the last few years. The attempts to directly construct AMG (algebraic multigrid) methods had some success, see [10, 2, 7]. Exploiting available efficient MG methods on auxiliary mesh for the same bilinear form led to efficient auxiliary mesh preconditioners to unstructured problems as shown in [8, 5]. The disadvantage of the latter approach is that one needs to re-discretize the given problem on a uniformly refined mesh. A computationally more attractive approach was recently announced by Hiptmair and Xu in [6]. Their method borrows the main tool from the auxiliary mesh preconditioners in [8] and [5], namely, the interpolation operator $\boldsymbol{\Pi}_h$ that maps functions from $\boldsymbol{H}(\mathrm{curl})$ into the respective Nédélec finite element space $\boldsymbol{V}_h$ as well as its transpose. The mapping $\boldsymbol{\Pi}_h$ involves computing line integrals over the edges of the tetrahedral elements for certain piecewise polynomial functions. The approach in [6] does not require re-meshing the domain on a related uniformly refined mesh. It requires implementing actions of $\boldsymbol{\Pi}_h$ as a mapping from $\boldsymbol{S}_h$, a related $\boldsymbol{H}^1$–conforming finite element space on the original (unstructured) triangulation, into the Nédélec space $\boldsymbol{V}_h$. In addition, the method requires explicit knowledge of the curl–free components of $\boldsymbol{V}_h$, which for Nédélec spaces, as is well–known, are gradients of a H^1–conforming (scalar) finite element space S_h. Those are represented as the range of a sparse matrix G_h mapping the degrees of freedom in S_h into degrees of freedom of $\boldsymbol{V}_h$. To compute G_h, one has to expand $\nabla\varphi_i$ in terms of the basis of $\boldsymbol{V}_h$, for any basis function $\varphi_i \in S_h$. For the lowest order Nédélec space, G_h is simply a "vertex"–to–"edge" mapping with entries -1 or 1. In summary, the tools required by the method in [6], are

(i) the original $\boldsymbol{H}(\mathrm{curl})$–problem in terms of a bilinear form $a(\cdot,\cdot)$, triangulation $\mathcal{T}_h$, finite element (Nédélec) space $\boldsymbol{V}_h$; the respective matrix $\boldsymbol{A}_h$ computed on the basis of $a(\cdot,\cdot)$ and $\boldsymbol{V}_h$.

Date: March 24, 2006–beginning; Today is May 22, 2006.

1991 *Mathematics Subject Classification.* 65F10, 65N20, 65N30.

This work was performed under the auspices of the U. S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

(ii) the scalar H^1–conforming finite element space S_h which gradients give the curl–free components of $\boldsymbol{V}_h$. This is typically provided by a matrix G_h that maps the degrees of freedom in S_h into the degrees of freedom in $\boldsymbol{V}_h$.
(iii) efficient preconditioner B_h for $G_h^T \boldsymbol{A}_h G_h$. Since, $G_h^T \boldsymbol{A}_h G_h$ corresponds to a finite element discretization of a 2nd order elliptic form, B_h can be for example an AMG preconditioner since $\mathcal{T}_h$ is generally unstructured mesh.
(iv) the symmetric Gauss–Seidel smoother Λ_h for $\boldsymbol{A}_h$.
(v) a (vector) $\boldsymbol{H}^1$–conforming f.e. space $\boldsymbol{S}_h$ associated with original triangulation $\mathcal{T}_h$ and the matrix representation of $\boldsymbol{\Pi}_h : \ \boldsymbol{S}_h \mapsto \boldsymbol{V}_h$ (the mapping that computes the standard interpolant in $\boldsymbol{V}_h$, which for the lowest Nédélec elements is based on computing line integrals from the tangential components of vector functions).
(vi) an AMG preconditioner for the matrix $\boldsymbol{\Pi}_h^T \boldsymbol{A}_h \boldsymbol{\Pi}_h$ or a perturbed version of it based on terms like $\delta_0 h^{-2} \|\mathbf{z}_h - \boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2$. The method in [6] originally suggested to use an optimal (A)MG preconditioner $\boldsymbol{B}_h$ based on a (vector) elliptic form $b(\cdot,\cdot)$ discretized using the space $\boldsymbol{S}_h$.

The resulting preconditioner (in its additive form) was based on the construction in [12]. It utilizes the three subspaces $\boldsymbol{V}_h$, $G_h S_h$, and $\boldsymbol{\Pi}_h \boldsymbol{S}_h$ of $\boldsymbol{V}_h$, and respective preconditioners Λ_h, $G_h B_h^{-1} G_h^T$ and $\boldsymbol{\Pi}_h \boldsymbol{B}_h^{-1} \boldsymbol{\Pi}_h^T$, in the following additive form,

$$\Lambda_h^{-1} + G_h B_h^{-1} G_h^T + \boldsymbol{\Pi}_h \boldsymbol{B}_h^{-1} \boldsymbol{\Pi}_h^T. \tag{1.1}$$

The purpose of the present report is to assess the quality of the approach in [6] by testing various options described in the items (i)–(vi) above.

The remainder of the report is organized as follows. In Section 2 we summarize the main idea of the approach in [6] in the present setting. In the following section 3 we introduce a perturbed form that is later used in the definition of the auxiliary space preconditioner, which we present in Section 4 in multiplicative form and outline a sketch of its analysis.

The main Section 5 contains an extensive set of numerical experiments. Our conclusion is that the best version in terms of performance is to build preconditioners on the basis of the original form; for example, as described in item (vi) it is best if one builds directly an AMG preconditioner for the form $\boldsymbol{\Pi}_h^T \boldsymbol{A}_h \boldsymbol{\Pi}_h$ and not on a spectrally equivalent $\boldsymbol{H}^1$ form $b(\cdot,\cdot)$.

2. Some consequences from a $\boldsymbol{H}(\text{curl})$–stable decompositions

In the present section we summarize the main ingredient used in [6] to derive and analyze the auxiliary space $\boldsymbol{H}(\text{curl})$–preconditioner.

Consider the lowest order Nédélec space $\boldsymbol{V}_h \subset \boldsymbol{H}_0(\text{curl})$. Let $\boldsymbol{\Pi}_h$ be the canonical interpolant into $\boldsymbol{V}_h$. Through the de Rham diagram (cf. e.g., [1]), $\boldsymbol{V}_h$ is related to the H^1 conforming scalar finite element space $S_h \subset H_0^1(\Omega)$ and its canonical interpolation operator I_h, i.e., one has $\boldsymbol{\Pi}_h \nabla \phi = \nabla I_h \phi$ for any $\phi \in S_h$. With the purpose of defining an auxiliary space preconditioner we will need the $\boldsymbol{H}_0^1$–conforming finite element space $\boldsymbol{S}_h = (S_h)^3$.

The result in [6] exploits the following decomposition (which we assume in what follows) of any $\mathbf{u}_h \in \boldsymbol{V}_h$,

$$\mathbf{u}_h = \mathbf{v}_h + \boldsymbol{\Pi}_h \mathbf{z}_h + \nabla \phi_h, \tag{2.1}$$

with

$$h^{-1} \|\mathbf{v}_h\|_0 + \|\mathbf{z}_h\|_1 + \|\nabla \phi_h\|_0 \leq C \; \|\mathbf{u}_h\|_{\boldsymbol{H}(\mathrm{curl})}. \tag{2.2}$$

Decomposition of the above type for the 2nd family of Nédélec elements (in place of $\boldsymbol{V}_h$, and $\boldsymbol{S}_h$ being quadratic conforming elements) has been used in [5].

The proof of (2.1)-(2.2) relies on the following regular decomposition of $H(\mathrm{curl})$ functions (cf., e.g., [9]),

$$\mathbf{u} = \mathbf{z} + \nabla \phi.$$

where $\mathbf{z} \in \boldsymbol{H}_0^1$ and $\phi \in H_0^1$ are stable components in the sense that

$$\|\mathbf{z}\|_1 + \|\phi\|_1 \leq C \; \|\mathbf{u}_h\|_{\boldsymbol{H}(\mathrm{curl})}.$$

Such decompositions are known to exist when the computational domain Ω has a connected boundary.

3. An $\boldsymbol{H}^1$–equivalent form

Here we study the case of lowest order Nédélec space $\boldsymbol{V}_h$.

Motivated by the result in [6], we introduce the following quadratic form on $\boldsymbol{S}_h$, for a given constant $\delta_0 > 0$,

$$(\overline{\boldsymbol{A}}_h \mathbf{z}_h, \; \mathbf{z}_h) \equiv \delta_0 h^{-2} \|\mathbf{z}_h - \boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2 + \|\operatorname{curl} \boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2 + \|\boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2. \tag{3.1}$$

We shall show that $(\overline{\boldsymbol{A}}_h \mathbf{z}_h, \; \mathbf{z}_h) \simeq \|\mathbf{z}_h\|_1^2$. We need to prove, that $\|\mathbf{z}_h\|_1^2 \leq C \; (\overline{\boldsymbol{A}}_h \mathbf{z}_h, \; \mathbf{z}_h)$. The latter is seen from the decomposition $\mathbf{z}_h = (\mathbf{z}_h - \boldsymbol{\Pi}_h \mathbf{z}_h) + \boldsymbol{\Pi}_h \mathbf{z}_h$. Taking element–wise gradients of both sides, one gets

$$\begin{aligned}
\|\mathbf{z}_h\|_1^2 &\leq 2 \sum_{\tau \in \mathcal{T}_h} \|\nabla(\mathbf{z}_h - \boldsymbol{\Pi}_h \mathbf{z}_h)\|_{0,\,\tau}^2 + 2 \sum_{\tau \in \mathcal{T}_h} \|\nabla \boldsymbol{\Pi}_h \mathbf{z}_h\|_{0,\,\tau}^2 \\
&\leq C \sum_{\tau \in \mathcal{T}_h} h^{-2} \; \|\mathbf{z}_h - \boldsymbol{\Pi}_h \mathbf{z}_h\|_{0,\,\tau} + 2 \sum_{\tau \in \mathcal{T}_h} \|\nabla \boldsymbol{\Pi}_h \mathbf{z}_h\|_{0,\,\tau}^2 \\
&\leq C \; h^{-2} \; \|\mathbf{z}_h - \boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2 + 2 \sum_{\tau \in \mathcal{T}_h} \|\nabla \boldsymbol{\Pi}_h \mathbf{z}_h\|_{0,\,\tau}^2.
\end{aligned}$$

Now, use the fact that on every element $\tau \in \mathcal{T}_h$ one has $\boldsymbol{\Pi}_h \mathbf{z}_h|_\tau = \mathbf{a}_\tau + \mathbf{b}_\tau \times \mathbf{x}$, which is a special type of linear polynomial. Therefore, $\nabla \boldsymbol{\Pi}_h \mathbf{z}_h|_\tau = \mathbf{b}_\tau \times \mathbf{1}$, where $\mathbf{1}$ is a constant matrix. Then,

$$\|\nabla \boldsymbol{\Pi}_h \mathbf{z}_h\|_{0,\,\tau}^2 \leq \mathrm{const} \; \|\mathbf{b}_\tau\|^2 \; |\tau|.$$

Here, $\|\mathbf{b}_\tau\|$ is the $\mathbb{R}^3$ Euclidean vector norm of the constant vector $\mathbf{b}_\tau$. Also, $|\tau|$ stands for the volume of the tetrahedron $\tau \in \mathcal{T}_h$. Next, notice that the element–wise curl of $\boldsymbol{\Pi}_h \mathbf{z}_h$ on τ equals $2\mathbf{b}_\tau$. Therefore,

$$\sum_{\tau \in \mathcal{T}_h} \|\nabla \boldsymbol{\Pi}_h \mathbf{z}_h\|_{0,\,\tau}^2 \leq \mathrm{const} \; \|\operatorname{curl} \boldsymbol{\Pi}_h \mathbf{z}_h\|_0^2.$$

In conclusion, we have the desired $\boldsymbol{H}_0^1$–coercivity estimate

$$\|\mathbf{z}_h\|_1^2 \leq C \; (\overline{\boldsymbol{A}}_h \mathbf{z}_h, \; \mathbf{z}_h), \quad \text{for all } \mathbf{z}_h \in \boldsymbol{S}_h.$$

This (together with the obvious $\boldsymbol{H}_0^1$ – boundedness of $\overline{\boldsymbol{A}}_h$) implies that efficient MG method, and AMG(e) for unstructured meshes, in particular, exist for solving problems involving $\overline{\boldsymbol{A}}_h$.

The purpose to introduce the penalty term was to ensure coercivity of the form $\overline{\boldsymbol{A}}_h$. Our numerical experiments though did not actually need that term; i.e., the methods based on $\delta_0 = 0$ worked very well.

4. The auxiliary space preconditioner

Let $\boldsymbol{A}_h$ be the $\boldsymbol{H}_0(\text{curl})$–form of our main interest, $\boldsymbol{A}_h(\mathbf{u}_h, \mathbf{v}_h) = (\operatorname{curl} \mathbf{u}_h,\ \operatorname{curl} \mathbf{v}_h) + (\mathbf{u}_h,\ \mathbf{v}_h)$. Define now the following "two–level" preconditioner using a "smoother" $\boldsymbol{\Lambda}_h$, and a correction based on the space, $\boldsymbol{\Pi}_h \boldsymbol{S}_h$. The form $\boldsymbol{A}_h$ restricted to the latter space can be represented as $\boldsymbol{\Pi}_h^T \boldsymbol{A}_h \boldsymbol{\Pi}_h$.

One first defines, (recall the definition (3.1) of $\overline{\boldsymbol{A}}_h$),

$$\overline{\boldsymbol{B}}_h = \begin{bmatrix} \boldsymbol{\Lambda}_h & 0 \\ \boldsymbol{\Pi}_h^T \boldsymbol{A}_h & I \end{bmatrix} \begin{bmatrix} (2\boldsymbol{\Lambda}_h - \boldsymbol{A}_h)^{-1} & 0 \\ 0 & \overline{\boldsymbol{A}}_h \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_h & \boldsymbol{A}_h \boldsymbol{\Pi}_h \\ 0 & I \end{bmatrix}.$$

The (multiplicative) auxiliary space preconditioner then reads

$$\boldsymbol{B}_h^{-1} = [I,\ \boldsymbol{\Pi}_h]\, \overline{\boldsymbol{B}}_h^{-1} [I,\ \boldsymbol{\Pi}_h]^T .$$

Assuming that the "smoother" is convergent, i.e., $(\boldsymbol{A}_h \mathbf{v}_h,\ \mathbf{v}_h) \leq (\boldsymbol{\Lambda}_h \mathbf{v}_h,\ \mathbf{v}_h)$, one has $(\boldsymbol{A}_h \mathbf{u}_h,\ \mathbf{u}_h) \leq (\boldsymbol{B}_h \mathbf{u}_h,\ \mathbf{u}_h)$, and the following identity holds (cf. e.g., [11]):

$$\begin{aligned}(\boldsymbol{B}_h \mathbf{u}_h,\ \mathbf{u}_h) = \inf_{\mathbf{z}_h \in \boldsymbol{S}_h} & ((\overline{\boldsymbol{A}}_h \mathbf{z}_h,\ \mathbf{z}_h) \\ & +((2\boldsymbol{\Lambda}_h - \boldsymbol{A}_h)^{-1}(\boldsymbol{\Lambda}_h(\mathbf{u}_h - \boldsymbol{\Pi}_h \mathbf{z}_h) + \boldsymbol{A}_h \boldsymbol{\Pi}_h \mathbf{z}_h),\ (\boldsymbol{\Lambda}_h(\mathbf{u}_h - \boldsymbol{\Pi}_h \mathbf{z}_h) + \boldsymbol{A}_h \boldsymbol{\Pi}_h \mathbf{z}_h))).\end{aligned}$$

An estimate from above is obtained if one uses the fact that $\boldsymbol{\Lambda}_h - \boldsymbol{A}_h$ is positive semi–definite. Then,

$$\begin{aligned}(\boldsymbol{B}_h \mathbf{u}_h,\ \mathbf{u}_h) \leq \inf_{\mathbf{z}_h \in \boldsymbol{S}_h} & ((\overline{\boldsymbol{A}}_h \mathbf{z}_h,\ \mathbf{z}_h) \\ & +(\boldsymbol{\Lambda}_h^{-1}(\boldsymbol{\Lambda}_h(\mathbf{u}_h - \boldsymbol{\Pi}_h \mathbf{z}_h) + \boldsymbol{A}_h \boldsymbol{\Pi}_h \mathbf{z}_h),\ (\boldsymbol{\Lambda}_h(\mathbf{u}_h - \boldsymbol{\Pi}_h \mathbf{z}_h) + \boldsymbol{A}_h \boldsymbol{\Pi}_h \mathbf{z}_h))).\end{aligned}$$

Using finally the Cauchy–Schwarz inequality, one arrives at,

$$(\boldsymbol{B}_h \mathbf{u}_h,\ \mathbf{u}_h) \leq \inf_{\mathbf{z}_h \in \boldsymbol{S}_h} (2(\boldsymbol{\Lambda}_h(\mathbf{u}_h - \boldsymbol{\Pi}_h \mathbf{z}_h),\ \mathbf{u}_h - \boldsymbol{\Pi}_h \mathbf{z}_h) + 2(\boldsymbol{A}_h \boldsymbol{\Pi}_h \mathbf{z}_h,\ \boldsymbol{\Pi}_h \mathbf{z}_h) + (\overline{\boldsymbol{A}}_h \mathbf{z}_h,\ \mathbf{z}_h)).$$

Now use $\mathbf{z}_h$ (its existence follows from the stable decomposition (2.1)–(2.2)) such that $\mathbf{u}_h - \boldsymbol{\Pi}_h \mathbf{z}_h = \mathbf{v}_h + \nabla \phi_h$, and the latter component can be efficiently handled by the "smoother" $\boldsymbol{\Lambda}_h$, i.e., $\|\mathbf{u}_h - \boldsymbol{\Pi}_h \mathbf{z}_h\|_{\boldsymbol{\Lambda}_h} \leq \eta\ \|\mathbf{u}_h\|_{\boldsymbol{A}_h}$. An appropriate smoother $\boldsymbol{\Lambda}_h$ is one based on Hiptmair's work [4]. In the present setting it can be constructed based on standard Gauss–Seidel smoothing on the original form $\boldsymbol{A}_h$ and a V–cycle applied to the form $\boldsymbol{A}_h(\nabla\phi,\ \nabla\phi)$ for $\phi \in S_h \subset H_0^1(\Omega)$. i.e., the original form restricted to the subspace ∇S_h.

It remains to bound the term $(\boldsymbol{A}_h \boldsymbol{\Pi}_h \mathbf{z}_h,\ \boldsymbol{\Pi}_h \mathbf{z}_h)$. It is obvious that

$$(\boldsymbol{A}_h \boldsymbol{\Pi}_h \mathbf{z}_h, \boldsymbol{\Pi}_h \mathbf{z}_h) \leq (\overline{\boldsymbol{A}}_h \mathbf{z}_h, \mathbf{z}_h) \simeq \|\mathbf{z}_h\|_1^2 \leq \eta\ \|\mathbf{u}_h\|^2_{\boldsymbol{H}(\text{curl})}. \tag{4.1}$$

The last estimate, completes the upper bound

$$(\boldsymbol{B}_h \mathbf{u}_h, \mathbf{u}_h) \leq C\ (\boldsymbol{A}_h \mathbf{u}_h, \mathbf{u}_h).$$

Remark 4.1. *In the definition of $\boldsymbol{B}_h$ we used the form $\overline{\boldsymbol{A}}_h$. However, since $\overline{\boldsymbol{A}}_h(\cdot,\ \cdot)$, is $\boldsymbol{H}_0^1$–equivalent on $\boldsymbol{S}_h$, it can be further replaced by any (A)MG method suitable for general unstructured meshes. Also, as proposed in [6], one can first replace $\overline{\boldsymbol{A}}_h$ with a $\boldsymbol{H}_0^1$–equivalent form, and then by a V–cycle based on that equivalent form. In the simplest case one can use three Laplacian–based V–cycles for each scalar component of $\mathbf{z}_h \in \boldsymbol{S}_h$.*

In conclusion, we showed that the auxiliary space preconditioner $\boldsymbol{B}_h$ with $\overline{\boldsymbol{A}}_h$ replaced by an (A)MG preconditioner for $\overline{\boldsymbol{A}}_h$, will give a spectrally equivalent auxiliary space preconditioner for the original form $\boldsymbol{A}_h$.

5. Numerical experiments

In this section we present results from numerical experiments with different versions of the preconditioner discussed in the $\boldsymbol{H}^1$-based auxiliary space preconditioner. The setup is given by items (i)–(vi) in the introduction. We did not find the use of the penalty term beneficial, so in all the tests we set $\delta_0 = 0$.

In our implementation we assume that we are given the following data:

- The Nédélec stiffness matrix $\boldsymbol{A}_h$.
- The discrete gradient matrix G_h.
- The coordinates of the vertices of the mesh (as three vectors): x, y, z.

Then, the matrix representation of $\boldsymbol{\Pi}_h$, can be computed by noting that

$$\boldsymbol{\Pi}_h = \begin{pmatrix} \Pi_h^x & \Pi_h^y & \Pi_h^z \end{pmatrix} ,$$

where each of the blocks has the same sparsity pattern as G_h. The two nonzero entries in the ith row of Π_h^x equal the ith entry of the vector $\frac{1}{2}\, G_h\, x$. Similarly Π_h^y and Π_h^z are determined from the vectors $G_h\, y$ and $G_h\, z$. Alternatively, the user can directly provide the vectors $G_h\, x$, $G_h\, y$ and $G_h\, z$, which correspond to the representations of the functions $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ in the basis of the Nédélec finite element space.

We used symmetric multiplicative Hiptmair "smoothers" which differ from the usual definition by the fact that the smoothing on the vertices is replaced by a multigrid V-cycle. The following preconditioners were considered:

(1) Multiplicative with AMG V-cycles in the subspaces (including the auxiliary space $\boldsymbol{S}_h$ and the Hiptmair "smoother").
(2) Additive with AMG V-cycles in the subspaces.
(3) Multiplicative with Poisson subspace solvers based on geometric multigrid (this is the method from [6] as discussed in Remark 4.1).
(4) Additive with Poisson subspace solvers based on geometric multigrid.
(5) Multiplicative with Poisson subspace solvers based on algebraic multigrid.

The AMG algorithm we used is a serial version of the BoomerAMG solver from the *hypre* library as described in [3].

We report the number of preconditioned conjugate gradient iterations with the above preconditioners and relative tolerance 10^{-6}, i.e. the iterations were stopped after the preconditioned residual norm was reduced by six orders of magnitude. In few of the tests we also tried the corresponding two-level methods (using exact solution in the subspaces) and listed the iteration counts in parenthesis following the V-cycle results.

5.1. **Constant coefficients.** First we consider problems with constant coefficients, i.e. we are preconditioning a discretization of the form

$$(\operatorname{curl}\mathbf{u}, \operatorname{curl}\mathbf{v}) + (\mathbf{u}, \mathbf{v})\,.$$

The results are listed in Tables 1–6, where the following notation was used: ℓ is the refinement level of the mesh, N is the size of the problem, and n_1 to n_5 give the iteration count for each of the two-level preconditioners (1) to (5). When available, the error in L^2 is also reported. Finally, few selected timings on a machine with 2.4GHz Xeon processor are presented in Table 7.

The experiments show that all considered solvers result in uniform and small number of iterations, which is in agreement with the theoretical results explained in the previous sections. One can also observe that the multilevel results are very close in terms of number of iterations to the two-level ones.

Note that the first two methods (based on the original form) appear to work the same, independently of how complicated the geometry is. This is particularly interesting in the case for the third problem, where the assumption that $\partial\Omega$ is connected (needed to establish the decomposition in Section 2) is violated. In contrast, the third and the forth methods (based on Poisson subspace solvers) consistently result in bigger number of iterations, and perform much worse on the third problem.

5.2. **Variable coefficients.** In Tables 8–12 we report some results from tests for the linear system arising from

$$(\alpha\operatorname{curl}\mathbf{u}, \operatorname{curl}\mathbf{v}) + (\beta\mathbf{u}, \mathbf{v})\,,$$

where α and β are piecewise constant coefficients. Note that this was not discussed and is not covered by the theory described in the preceding sections. The modifications to the preconditioners (1)-(5) are straightforward, for example the Poisson-based preconditioners assemble matrices corresponding to the bilinear forms $(\beta\nabla u, \nabla v)$ and $(\alpha\nabla\mathbf{u}, \nabla\mathbf{v}) + (\beta\mathbf{u}, \mathbf{v})$. In the present case we concentrated only on the multiplicative AMG methods.

For problems with simple jumps (Tables 8–11), we observe stable number of iterations both with respect to the mesh size and the magnitude of the jumps. In particular the methods do quite well on the problem illustrated in Table 10, which was reported to be problematic for geometric multigrid in [4]. As before, the method based on the original form outperforms the one based on a AMG Poisson subspace solver.

In Table 12 we consider a complicated problem having a lot of jumps both in α and β. This turned out to be a challenging tasks for all V-cycle methods. However, the two-level results for n_1 seem to indicate that a significant improvement can be achieved if a better algebraic solver is used for the original form.

5.3. **Singular problems.** Tables 13–15 present results for the problem corresponding to $\alpha = 1$, $\beta = 0$, i.e., to the bilinear form $(\operatorname{curl}\mathbf{u}, \operatorname{curl}\mathbf{v})$. In this case the matrix is singular, and the right-hand side, as well as the solution, belong to the space of discretely divergence free vectors (the kernel of G_h^T). Since $\beta = 0$, the solvers were modified to skip the correction in the space $G_h S_h$. This leads to a simpler preconditioner which in additive form reads

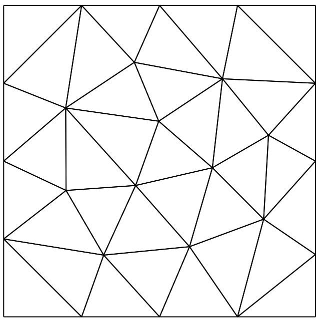
$$\Lambda_h^{-1} + \boldsymbol{\Pi}_h \boldsymbol{B}_h^{-1} \boldsymbol{\Pi}_h^T. \tag{5.1}$$

The results in Tables 13–14 are quite satisfactory and comparable to those from Tables 3 and 6. This should not be too surprising, since the terms in (5.1) are the ones that are supposed to take care of the component of the error not reduced by $G_h B_h^{-1} G_h^T$. These are precisely the discretely divergence–free vector fields.

In Table 15 we also consider the important practical case when β is zero only in part of the region. For this test we used a preconditioner based on (1.1) instead of (5.1). The numbers of iterations are comparable to those from Table 8.

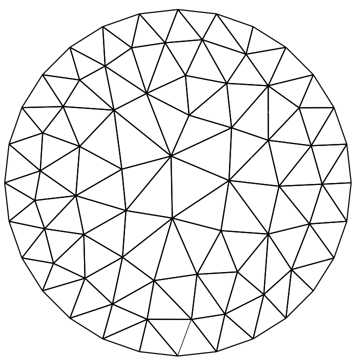
5.4. **Anisotropic problems.** The final set of experiments is shown in Tables 16–17, where $\alpha = \beta = 1$ and we consider two anisotropically refined meshes.

The results in Table 16, show significantly better, and scalable performance for the method based on the original form compared to the AMG Poisson subspace solver. The problem from Table 17 is much more challenging, but the number of iterations of our first preconditioner is still reasonable (if one takes into account that severe anisotropy is problematic for AMG).



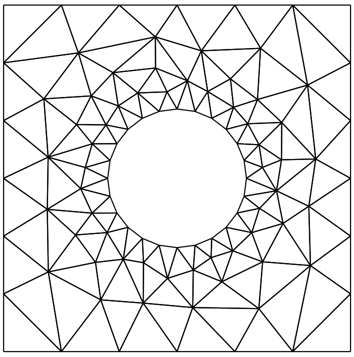
ℓ	N	n_1	n_2	n_3	n_4	$\|e\|_{L^2}$
2	896	4 (3)	9 (9)	10 (9)	16 (15)	0.011898
3	3520	4 (3)	10 (9)	11 (10)	17 (16)	0.005953
4	13952	4 (3)	10 (9)	12 (11)	18 (15)	0.002977
5	55552	4 (3)	10 (9)	13 (11)	18 (16)	0.001489
6	221696	4 (3)	10 (8)	13 (11)	18 (16)	0.000744
7	885760	5	10	13	18	0.000372
8	3540992	5	11	13	19	0.000186

TABLE 1. Initial mesh and numerical results for the problem on a square.



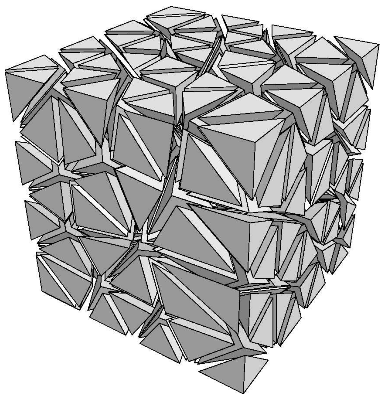
ℓ	N	n_1	n_2	n_3	n_4
1	736	4 (2)	9 (8)	7 (7)	11 (11)
2	2888	4 (3)	10 (9)	7 (7)	12 (11)
3	11440	4 (3)	10 (9)	7 (7)	12 (11)
4	45536	5 (3)	11 (9)	7 (7)	12 (11)
5	181696	5 (3)	11 (8)	8 (7)	12 (11)
6	725888	5	11	8	12
7	2901760	5	12	8	11

TABLE 2. Initial mesh and numerical results for the problem on a disk.



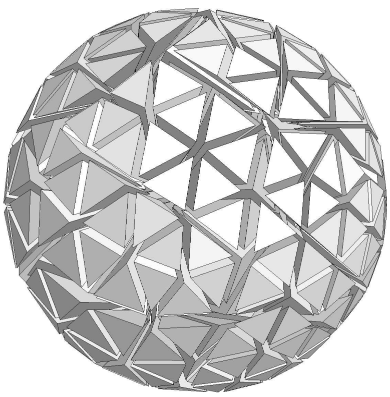
ℓ	N	n_1	n_2	n_3	n_4
2	972	6 (3)	11 (9)	21 (20)	33 (31)
3	14976	6 (3)	12 (9)	23 (21)	33 (31)
4	59520	7 (3)	12 (9)	23 (17)	35 (23)
5	237312	6	13	24	35
6	947712	7	13	25	35
7	3787776	7	14	25	35

Table 3. Initial mesh and numerical results for the problem on a square with circular hole.



ℓ	N	n_1	n_2	n_3	n_4	$\|e\|_{L^2}$
0	722	3 (3)	9 (7)	6 (6)	11 (11)	0.6777
1	5074	4 (3)	10 (9)	9 (9)	16 (15)	0.3776
2	37940	5 (4)	11 (10)	12 (11)	20 (19)	0.2152
3	293224	5 (4)	11 (10)	14 (12)	22 (20)	0.1096
4	2305232	5	11	15	23	0.0549

Table 4. Initial mesh and numerical results for the problem on a cube.

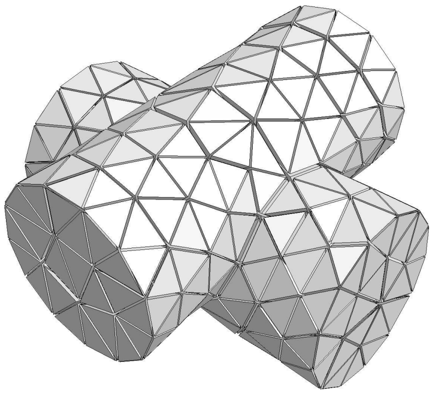


ℓ	N	n_1	n_2	n_3	n_4
0	704	3 (3)	9 (7)	5 (5)	9 (9)
1	4669	4 (3)	10 (9)	7 (7)	13 (12)
2	37940	5 (4)	12 (10)	8 (8)	15 (14)
3	255700	6 (5)	13 (12)	9 (8)	15 (14)
4	1990184	7	14	10	16

Table 5. Initial mesh and numerical results for the problem on a ball.

References

[1] D. N. Arnold, R. S. Falk, and R. Winther. Finite element exterior calculus; homological techniques, and applications, University of Minnesota, IMA Preprint Series # 2094(February 2006).



ℓ	N	n_1	n_2	n_3	n_4
0	1197	3 (3)	10 (8)	7 (7)	13 (13)
1	8248	4 (4)	11 (10)	10 (9)	17 (16)
2	60940	5 (5)	12 (11)	12 (11)	19 (18)
3	467880	6 (5)	12 (11)	13 (12)	21 (19)
4	3665552	6	13	14	22

TABLE 6. Initial mesh and numerical results for the problem on a union of two cylinders.

N	t_R	$t_3(n_3)$	$t_4(n_4)$
3540992	7s	128s (13)	122s (19)
2901760	6s	67s (8)	60s (11)
3787776	7s	255s (25)	239s (35)
2305232	9s	146s (16)	141s (25)
1990184	7s	79s (10)	82s (17)

TABLE 7. Times for setup and solution of few selected problems on a machine with 2.4GHz Xeon processor.

[2] P. Bochev, C. Garasi, J. Hu, A. Robinson and R. Tuminaro, An improved algebraic multigrid method for solving Maxwell's equations, *SIAM J. Sci. Comp.*, 25:623-642, 2003.

[3] V. E. HENSON AND U. M. YANG, BoomerAMG: A Parallel Algebraic Multigrid Solver and Preconditioner. *Applied Numerical Mathematics*, 41:155–177, 2002.

[4] R. Hiptmair. Multigrid method for Maxwell's equations. *SIAM J. Numer. Anal.*, 36(1):204–225, 1999.

[5] R. Hiptmair, G. Widmer, and J. Zou, Auxiliary space preconditioning in $\boldsymbol{H}$(curl), *Numerische Mathematik*, 103(3):435–459, 2006.

[6] R. Hiptmair and J. Xu, Nodal auxiliary space preconditioning in $\boldsymbol{H}$(curl) and $\boldsymbol{H}$(div) spaces, *Research Report No. 2006-09*, Seminar für Angewandte Mathematik, Eidgenössische Technische Hochschule, CH-8092 Zürich, Switzerland, May 2006.

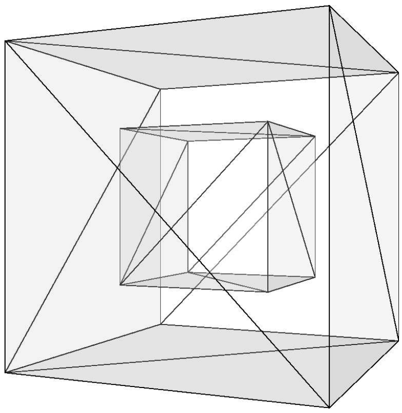
[7] J. Jones and B. Lee, A multigrid method for variable coefficient Maxwell's equations, *SIAM J. Sci. Comp.*, 27:1689–1708, 2006.

[8] Tz. V. Kolev, J. E. Pasciak and P. S. Vassilevski, $\boldsymbol{H}$(curl) auxiliary mesh preconditioning, *in preparation*

[9] J. E. Pasciak and J. Zhao, Overlapping Schwarz methods in $\boldsymbol{H}$(curl) on polyhedral domain, *East West J. Numer. Anal.*, 10:221-234, 2002.

[10] S. Reitzinger and J. Schöberl, An algebraic multigrid method for finite element discretization with edge elements, *Num. Lin. Alg. Appl.*, 9:215–235, 2002.

[11] P. S. Vassilevski, A block–factorization (algebraic) formulation of multigrid and Schwarz methods, *East–West J. Numer. Math.* **6**(1998), pp. 65–79.



ℓ	N	p								
		-8	-4	-2	-1	0	1	2	4	8
n_1 for $\alpha = 1, \beta \in \{1, 10^p\}$										
1	485	4	4	4	4	4	4	5	6	6
2	3674	6	6	6	6	6	6	7	8	9
3	28692	7	7	7	7	7	7	7	9	10
4	226984	7	7	7	7	7	7	8	10	10
5	1806160	8	8	8	8	8	8	8	10	10
n_1 for $\beta = 1, \alpha \in \{1, 10^p\}$										
1	485	5	5	5	5	4	4	5	5	5
2	3674	6	6	6	6	6	5	6	7	7
3	28692	7	7	7	7	7	7	7	7	7
4	226984	7	8	7	7	7	7	8	8	8
5	1806160	8	8	8	8	8	8	8	8	8

Table 8. Initial mesh and numerical results for the problem on a cube with α and β having different values inside and outside the interior cube. Multiplicative preconditioner with AMG V-cycles in the subspaces.

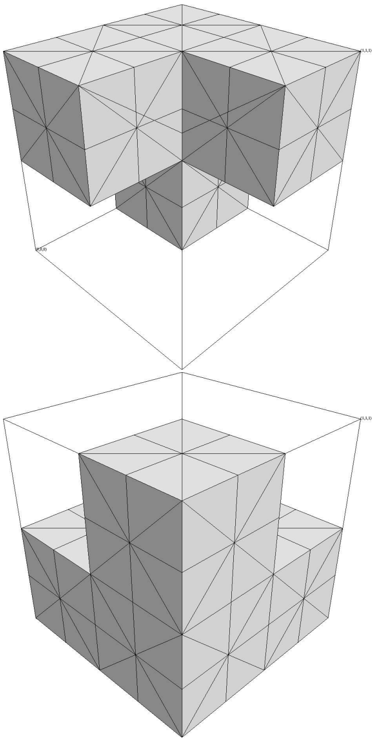
ℓ	N	p								
		-8	-4	-2	-1	0	1	2	4	8
n_5 for $\alpha = 1, \beta \in \{1, 10^p\}$										
1	485	6 (6)	6 (6)	6 (6)	6 (6)	5 (6)	6 (6)	7 (7)	8 (8)	8 (8)
2	3674	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	9 (9)	14 (14)	14 (14)
3	28692	9 (9)	9 (9)	9 (9)	9 (9)	9 (9)	9 (9)	11 (11)	20 (19)	21 (20)
4	226984	11 (10)	11 (10)	11 (10)	11 (10)	11 (11)	11 (11)	12 (12)	23 (20)	26 (24)
5	1806160	12	12	12	12	12	12	13	21	26
n_5 for $\beta = 1, \alpha \in \{1, 10^p\}$										
1	485	6 (6)	6 (6)	6 (6)	6 (6)	5 (6)	6 (6)	6 (6)	6 (6)	6 (6)
2	3674	8 (7)	8 (7)	8 (7)	7 (7)	7 (7)	9 (9)	11 (10)	11 (10)	10 (10)
3	28692	9 (9)	9 (9)	9 (9)	9 (9)	9 (9)	13 (12)	16 (15)	16 (16)	15 (15)
4	226984	11 (11)	11 (11)	11 (11)	11 (11)	11 (10)	15 (14)	19 (17)	21 (19)	19 (18)
5	1806160	12	12	13	13	12	16	21	23	22

Table 9. Numerical results for the problem from Table 8 using multiplicative preconditioner with Poisson subspace solvers based on algebraic multigrid.

[12] J. Xu. The auxiliary space method and optimal preconditioning techniques for unstructured grids, *Computing*, 56:215–235, 1996.

Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551, U.S.A.

E-mail address: tzanio@llnl.gov, panayot@llnl.gov

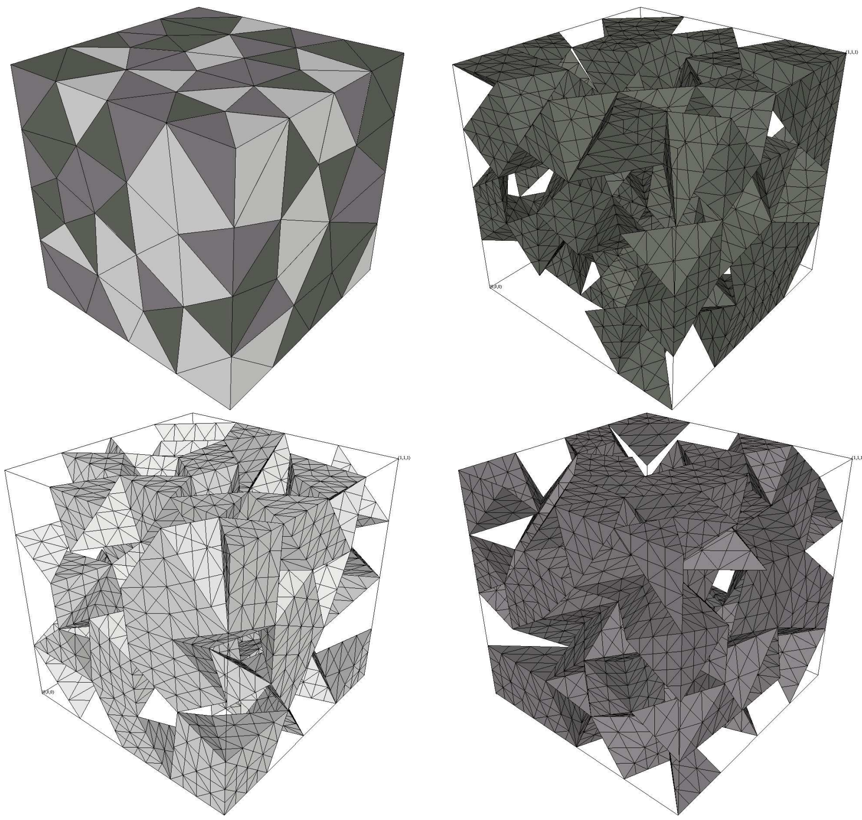


ℓ	N	p								
		-8	-4	-2	-1	0	1	2	4	8
n_1 for $\alpha = 1, \beta \in \{1, 10^p\}$										
1	716	3	3	3	3	3	4	4	4	4
2	5080	4	4	4	4	4	4	5	6	6
3	38192	5	5	5	5	5	5	5	6	6
4	296032	5	5	5	5	5	5	6	6	6
5	2330816	5	5	5	5	5	6	6	6	6
n_1 for $\beta = 1, \alpha \in \{1, 10^p\}$										
1	716	6	6	5	4	3	4	4	4	4
2	5080	6	6	6	5	4	5	5	5	5
3	38192	7	7	7	5	5	5	6	6	6
4	296032	8	8	7	6	5	6	6	6	6
5	2330816	8	9	7	6	5	6	6	6	6

TABLE 10. Numerical results for the problem on a cube with α and β having different values in the shown regions (cf. [4]). Multiplicative preconditioner with AMG V-cycles in the subspaces.

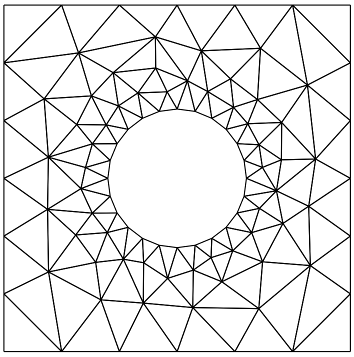
ℓ	N	p								
		-8	-4	-2	-1	0	1	2	4	8
n_5 for $\alpha = 1, \beta \in \{1, 10^p\}$										
1	716	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	6 (6)	6 (6)	5 (5)	5 (5)
2	5080	10 (10)	10 (10)	10 (10)	10 (10)	10 (10)	10 (10)	9 (9)	9 (9)	9 (9)
3	38192	11 (11)	11 (11)	11 (11)	11 (11)	11 (11)	11 (11)	10 (10)	11 (11)	12 (11)
4	296032	12 (12)	12 (12)	12 (12)	12 (12)	12 (12)	12 (12)	12 (12)	13 (13)	14 (13)
5	2330816	14	14	14	14	14	13	13	14	15
n_5 for $\beta = 1, \alpha \in \{1, 10^p\}$										
1	716	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	6 (6)
2	5080	10 (9)	9 (9)	10 (10)	10 (10)	10 (10)	10 (10)	10 (10)	10 (10)	9 (9)
3	38192	11 (11)	11 (11)	12 (12)	12 (12)	11 (11)	12 (12)	12 (12)	12 (12)	12 (12)
4	296032	13 (13)	13 (13)	14 (14)	14 (14)	12 (12)	15 (14)	15 (15)	15 (15)	14 (14)
5	2330816	15	15	16	16	14	16	17	17	17

TABLE 11. Numerical results for the problem from Table 10 using multiplicative preconditioner with Poisson subspace solvers based on algebraic multigrid.



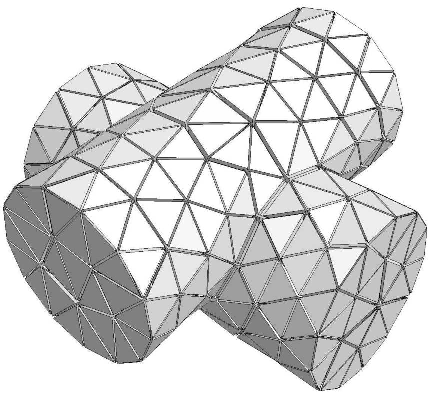
ℓ	N	p								
		-8	-4	-2	-1	0	1	2	4	8
n_1 for $\alpha \in \{10^p, 1, 1\}, \beta \in \{1, 1, 10^{-p}\}$										
0	722	3	3	5 (5)	4 (4)	3 (3)	5 (4)	14 (11)	48	69
1	5074	10	9	8 (7)	6 (5)	4 (3)	6 (5)	16 (7)	60	102
2	37940	17	14	10 (9)	7 (6)	5 (4)	7 (5)	15 (7)	80	160
3	296032	19	15	11 (9)	7 (6)	5 (4)	7 (5)	16 (7)	91	196
n_3 for $\alpha \in \{10^p, 1, 1\}, \beta \in \{1, 1, 10^{-p}\}$										
0	722	3	3	5	6	6	8	20	48	64
1	5074	10	10	10	11	9	13	30	89	120
2	37940	20	17	16	16	12	17	41	134	208
3	296032	27	23	20	20	14	21	50	181	252
n_5 for $\alpha \in \{10^p, 1, 1\}, \beta \in \{1, 1, 10^{-p}\}$										
0	722	3	3	5	6	6	8	20	60	91
1	5074	10	10	10	11	9	13	29	100	158
2	37940	20	16	14	15	11	16	37	135	219
3	296032	24	22	20	19	12	20	44	172	286

TABLE 12. Numerical results for the problem on a cube with α and β having different values in the three randomly selected groups of coarse elements shown above. Iteration counts for the considered multiplicative preconditioners.



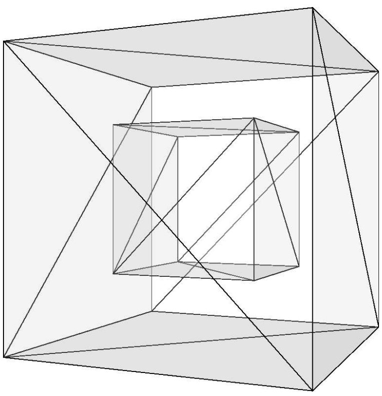
ℓ	N	n_1	n_2	n_3	n_4
2	972	7	12	19	28
3	14976	6	12	19	28
4	59520	7	12	19	28
5	237312	7	12	20	29
6	947712	7	11	20	29
7	3787776	7	12	21	29

TABLE 13. Initial mesh and numerical results for the singular problem on a square with circular hole.



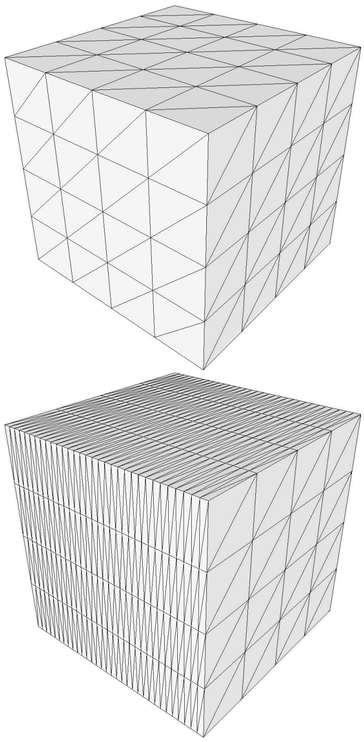
ℓ	N	n_1	n_2	n_3	n_4
0	1197	5	11	9	17
1	8248	6	13	12	19
2	60940	7	15	13	22
3	467880	7	15	14	23
4	3665552	8	15	15	23

TABLE 14. Initial mesh and numerical results for the singular problem on a union of two cylinders.



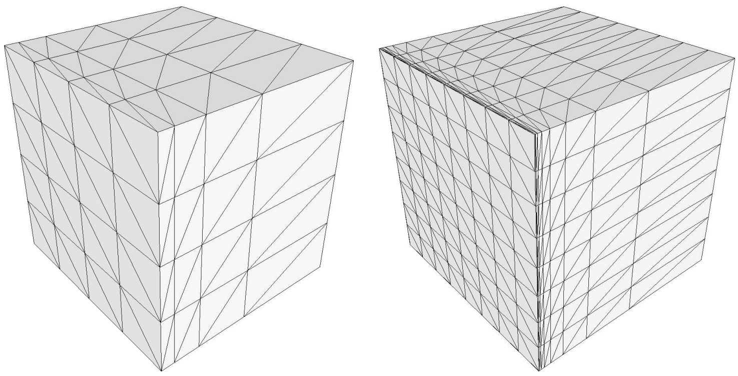
ℓ	N	p								
		-8	-4	-2	-1	0	1	2	4	8
n_1 for $\alpha = 1, \beta \in \{0, 10^p\}$										
1	485	2	3	3	2	4	2	3	3	3
2	3674	5	5	5	6	5	6	6	6	7
3	28692	8	7	8	7	7	8	8	10	10
4	226984	7	7	7	7	7	9	8	9	9
5	1806160	8	8	8	8	8	8	9	10	11

TABLE 15. Initial mesh and numerical results for the problem on a cube with $\beta = 0$ outside the interior cube. Multiplicative preconditioner with AMG V-cycles in the subspaces.



ℓ	h_y/h_x	N	n_1	n_5
0	2^0	604	3	4
1	2^1	1152	3	7
2	2^2	2248	5	11
3	2^3	4440	8	18
4	2^4	8824	11	33
5	2^5	17592	16	60
6	2^6	35128	20	115
7	2^7	70200	22	217
8	2^8	140344	23	406
9	2^9	280632	24	801
10	2^{10}	561208	23	
11	2^{11}	1122360	23	
12	2^{12}	2244664	22	

Table 16. Numerical results for the problem on a cube with a mesh that is refined only in the x-direction (the ratio of the mesh sizes in y and x directions is given in the second column). The meshes corresponding to $\ell = 0$ and $\ell = 3$ are shown on the left. Multiplicative preconditioners using AMG.



ℓ	h_x/h_y	N	n_1
0	2^2	604	3
1	2^5	4184	5
2	2^{12}	31024	8
3	2^{27}	238688	11
4	2^{58}	1872064	18

Table 17. Numerical results for the problem on a cube with exponential local refinement in the y-direction. The x and z directions are refined uniformly and the ratio of the mesh size in x-direction and the minimal mesh size in y-direction is given in the second column. The meshes corresponding to $\ell = 0$ and $\ell = 1$ are shown on the left.